

REMARKS/ARGUMENTS

The present amendment is submitted in response to the Office Action received from the United States Patent Office dated March 5, 2008. In the Office Action, the Patent Office requested that the title be changed. Moreover, in the Office Action, the Patent Office rejected Claims 1-14 and 31-32 under 35 U.S.C. §103(a) as being unpatentable over *Advanced Compiler Design & Implementation*, Steven S. Muchnick, August 19, 1997 (herein after *Advanced*) and *Kawahito et al.* (United States Patent Number: 6,519,765). The Patent Office further rejected Claims 15-30 and 33-36 under 35 U.S.C. §102(b) as being anticipated by *Fast Static Analysis of C++ Virtual Function Calls*, David F. Bacon et al., ACM, 1996, pages 324-341. Additionally, the Patent Office rejected Claims 37-42 under 35 U.S.C. §102(b) as being anticipated by *Advanced*. Still further, the Patent Office rejected Claims 43-44 under 35 U.S.C. §102(b) as being anticipated by *Reducing Interprocess Communication Overhead in Concurrent Programs*, Erick Stenman et al, ACM, 2002, 6 pages (herein after referred to as *Reducing*). The Patent Office further rejected Claims 45 and 46 under 35 U.S.C. §102(b) as being anticipated by *Exception Analysis for Non-Strict Languages*, Kevin Glynn, ACM, 2002, pages 98-109 (herein after referred to as *Exception*). Finally, the Patent Office rejected Claims 47 and 48 under 35 U.S.C. §102(b) as being anticipated by *McKinsey* (United States Patent Number: 6,446,258).

In response to the Office Action, Applicant has amended the claims to overcome the rejections. Applicant respectfully submits that the amendments to the claims and the explanations below overcome the rejections to the claims. Applicant submits that all of the claims are now in condition for allowance. Notice to that effect is requested.

The Patent Office states that the title of the invention is not descriptive. The Patent Office states that a new title is required that is clearly indicative of the invention to which the claims are directed. The Patent Office states that words like “Method” and “system” should be removed. In response to same, applicant has changed the title to “Program Transformation Using Flow-Sensitive Type Constraint Analysis.”

The Patent Office rejected Claims 1-14 and 31-32 under 35 U.S.C. §103(a) as being unpatentable over *Advanced* in view of *Kawahito et al.* The Patent Office states that *Advanced*

teaches the common techniques of compiler theory and optimization and *Kawahito et al.* teaches data flow analysis. Therefore it would have been obvious to one of ordinary skill at the time of invention to combine the teachings because optimized software is more efficient. As to Claim 1, the Patent Office states that *Advanced* teaches a method for analyzing a program, comprising: determining a set of functions required by the program by performing local type constraint analysis at intermediate language instruction level and a call path that may reach a function containing such instruction. The Patent Office further states that *Kawahito et al.* teaches the optimization at runtime.

Further the Patent Office states that a computer-readable medium storing computer-executable process steps of a process for analyzing a program, comprising: determining a set of functions required by the program by performing local type constraint analysis at intermediate language instruction level and a call is taught.

Amended Claim 1 requires a method for analyzing a program, comprising: determining a set of functions required by the program by performing local type constraint analysis at intermediate language instruction level to determine which functions have the potential of being executed; and determining a call path that may reach a function containing such instruction.

Amended claim 8 requires a computer-readable medium storing computer-executable process steps of a process for analyzing a program, comprising: determining a set of functions required by the program by performing local type constraint analysis at intermediate language instruction level to determine which functions have the potential of being executed and determining a call path that may reach a function containing such instruction.

Neither *Advanced Compiler Design & Implementation* or *Kawahito et al.* teaches or suggest determining a set of functions required by the program by performing local type constraint analysis at intermediate language instruction level to determine which functions have the potential of being executed and determining a call path that may reach a function containing such instruction as required by Claim 1 and 8.

It is further submitted that the question under §103 is whether the totality of the art would collectively suggest the claimed invention to one of ordinary skill in this art. *In re Simon*, 461 F.2d 1387, 174 USPQ 114 (CCPA 1972).

That elements, even distinguishing elements, are disclosed in the art is alone insufficient. It is common to find elements somewhere in the art. Moreover, most if not all elements perform their ordained and expected functions. The test is whether the invention as a whole, in light of the teaching of the reference, would have been obvious to one of ordinary skill in the art at the time the invention was made. *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 220 USPQ 193 (Fed. Cir. 1983).

It is insufficient that the art disclosed components of Applicants' invention. A teaching, suggestion, or incentive must exist to make the combination made by Applicants. *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed. Cir. 1988).

In view of the foregoing remarks and amendments, the rejection of Claims 1-14 and 31-32 under U.S.C. §103(a) as being unpatentable over *Advanced Compiler Design & Implementation* and *Kawahito*, have been overcome. Notice to that effect is requested.

The Patent Office rejected Claims 15-30 and 33-36 under 35 U.S.C. §102(b) as being anticipated by *Fast Static Analysis of C++ Virtual Function Calls*. The Patent Office states that the prior art reference anticipates Claim 15 in that the prior art discloses a method for analyzing a program, comprising: determining an object type that may exist at an execution point of the program, wherein this enables determination of a possible virtual function that may be called. The Patent Office states that the prior art also discloses a computer-readable medium storing computer-executable process steps of a process for analyzing a program, comprising: determining an object type that may exist at an execution point of the program, wherein this enables determination of possible virtual functions that may be called. The Patent Office also states that the prior art discloses a method for building an application, comprising: receiving source code instruction; determining optimum code requirement; and compiling native processor image as taught in Claim 25 and 28.

Applicant has amended Claims 15, 20, 25 and 28 in response to the above identified rejection. More specifically, the prior art has tried to address the problem of eliminating, but not all unused virtual functions. *Fast Static* discloses partially eliminating non-virtual functions, but does not eliminate all of these non-virtual function which maintains the problem of having dead code that still remains.

Fast Static teaches the ability to improve C++ programs resolving virtual function calls, thereby reducing compiled code size and reducing program complexity so as to improve human and automated program understanding and analysis. The Static study found that using their system, 71 percent of the virtual function calls were resolved. However, almost 30 percent of the dead code still remained and the system still was hampered by unresolved virtual function calls which obviously slow processing speed and take up valuable memory.

Amended Claim 15 teaches a method for analyzing a program, comprising: determining an object type that may exist at an execution point of the program and evaluating all possible object types that are created at every instruction of a program and carrying the object types through a stack evaluation, wherein this enables determination of a possible virtual function that may be called.

Amended Claim 20 teaches a computer-readable medium storing computer-executable process steps of a process for analyzing a program, comprising: determining an object type that may exist at an execution point of the program and evaluating all possible object types that are created at every instruction of a program and carrying the object types through a stack evaluation, wherein this enables determination of possible virtual functions that may be called.

Amended Claim 25 teaches a method for building an application, comprising: intermediate language receiving instructions and interpreting and analyzing same; determining optimum code requirement; and compiling native processor functions comprising native functions that return a declared type; native functions that return a set of types and return functions that vary according to input parameters.

Amended Claim 28 teaches a computer-readable medium storing computer-executable process steps of a process for building an application, comprising: intermediate language

receiving instructions and interpreting and analyzing same; determining optimum code requirement; and compiling native processor functions comprising native functions that return a declared type; native functions that return a set of types and return functions that vary according to input parameters.

Fast Static does not teach or suggest determining an object type that may exist at an execution point of the program and evaluating all possible object types that are created at every instruction of a program and carrying the object types through a stack evaluation, wherein this enables determination of a possible virtual function that may be called as required by Claim 15 and 20. Further *Fast Static* does not teach or suggest intermediate language receiving instructions and interpreting and analyzing same, determining optimum code requirement and compiling native processor functions comprising native functions that return a declared type; native functions that return a set of types and return functions that vary according to input parameters as required by Claims 25 and 28.

Under 35 U.S.C. §102(b), anticipation requires that a single reference disclose each and every element of Applicant's claimed invention. *Akzo N.V. v. U.S. International Trade Commission*, 808 F.2d 1471, 1479, 1 USPQ 2d 1241, 1245 (Fed. Cir. 1986).

Moreover, anticipation is not shown even if the differences between the claims and the reference are "insubstantial" and one skilled in the art could supply the missing elements. *Structure Rubber Products Co. v. Park Rubber Co.*, 749 F.2d. 707, 716, 223 USPQ 1264, 1270 (Fed. Cir. 1984).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (MPEP § 2131).

In view of the foregoing remarks and amendments, the rejection of Claims 15-30 and 33-36 under 35 U.S.C. §102(b) as being anticipated by *Fast Static Analysis of C++ Virtual Function Calls* has been overcome and should be withdrawn. Notice to that effect is requested.

The Patent Office further rejected Claims 37-42 under 35 U.S.C. §102(b) as being anticipated by *Advanced Compiler Design & Implementations*. The Patent Office states that

Advanced anticipates the method for determining variable size in a program, comprising: tracking variable size; and reducing variable size for program execution. Further the Patent Office states that the prior art reference teaches a computer-readable medium storing computer-executable process steps of a process for determining variable size in a program, comprising: tracking variable size; and reducing variable size for program execution.

Applicant has amended Claims 37 and 40 to overcome the rejection. More specifically, Amended Claim 37 requires a method for determining variable size in a program, comprising: analyzing program function calls recursively and tracking variable size; reducing variable size of program function calls for program execution.

Amended Claim 40 requires a computer-readable medium storing computer-executable process steps of a process for determining variable size in a program, comprising: analyzing program function calls recursively and tracking variable size; reducing variable size of program function calls for program execution.

Advanced Compiler does not describe a method for determining variable size in a program, comprising: analyzing program function calls recursively and tracking variable size; reducing variable size of program function calls for program execution as required by Claim 37 and 40.

Under 35 U.S.C. §102(b), anticipation requires that a single reference disclose each and every element of Applicant's claimed invention. *Akzo N.V. v. U.S. International Trade Commission*, 808 F.2d 1471, 1479, 1 USPQ 2d 1241, 1245 (Fed. Cir. 1986).

Moreover, anticipation is not shown even if the differences between the claims and the reference are "insubstantial" and one skilled in the art could supply the missing elements. *Structure Rubber Products Co. v. Park Rubber Co.*, 749 F.2d. 707, 716, 223 USPQ 1264, 1270 (Fed. Cir. 1984).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (MPEP § 2131).

In view of the foregoing remarks and amendments, the rejection of Claims 37-42 under 35 U.S.C. §102(b) as being anticipated by *Advanced Compiler Design & Implementations* has been overcome and should be withdrawn. Notice to that effect is requested.

The Patent Office further rejected Claims 43 and 44 under 35 U.S.C. §102(b) as being anticipated by “*On Reducing Interprocess Communication Overhead in Concurrent Programs.*” The Patent Office states that the prior art teaches a method for reducing empty function calls in a program comprising: determining if a call is made to an empty function; and removing a call that is made to an empty function. Further the Patent Office states that the prior art teaches a computer-readable medium storing computer-executable process steps of a process for reducing empty function calls in a program, comprising: determining if a call is made to an empty function; and removing a call that is made to an empty function.

The prior art reference teaches increasing the performance of highly concurrent programs in general and more specifically, Erlang programs. The prior art range from simple implementation tricks that reduce communication latency to more thorough code that rewrites guided by inlining across process boundaries. The prior art also discusses the impact of different heap architectures on interprocess communication in general and optimizations of same.

Amended Claim 43 requires a method for reducing empty function calls in a program, comprising: determining if a call is made to an empty function; and removing code that creates exceptions where exceptions are not handled; and removing code that checks values where the values can be determined in advance.

Amended Claim 44 requires a computer-readable medium storing computer-executable process steps of a process for reducing empty function calls in a program, comprising: determining if a call is made to an empty function; and removing code that creates exceptions where exceptions are not handled; and removing code that checks values where the values can be determined in advance.

The prior art does not teach or suggest a method for reducing empty function calls in a program, comprising: determining if a call is made to an empty function; and removing code that

creates exceptions where exceptions are not handled; and removing code that checks values where the values can be determined in advance as required by Claims 43 and 44.

Under 35 U.S.C. §102(b), anticipation requires that a single reference disclose each and every element of Applicant's claimed invention. *Akzo N.V. v. U.S. International Trade Commission*, 808 F.2d 1471, 1479, 1 USPQ 2d 1241, 1245 (Fed. Cir. 1986).

Moreover, anticipation is not shown even if the differences between the claims and the reference are "insubstantial" and one skilled in the art could supply the missing elements. *Structure Rubber Products Co. v. Park Rubber Co.*, 749 F.2d. 707, 716, 223 USPQ 1264, 1270 (Fed. Cir. 1984).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (MPEP § 2131).

In view of the foregoing remarks and amendments, the rejection of Claims 43 and 44 under 35 U.S.C. §102(b) as being anticipated by *On Reducing Interprocess Communication Overhead in Concurrent Programs* has been overcome and should be withdrawn. Notice to that effect is requested.

Further, the Patent Office rejected Claims 45 and 46 under 35 U.S.C. §102(b) as being anticipated by "Exception Analysis for Non-Strict Languages." The Patent Office states that the prior art teaches a method for reducing throw instruction without exception handlers in a program, comprising: determining if there are any throw instructions without exception handlers; and removing throw instructions without exception handlers.

The prior art references teaches a first exception analysis for a non-strict language. The prior art reference augments a simple type functional language with exceptions, and shows that a type based inference system can be shown to detect uncaught exceptions.

Applicant respectfully submits that the rejection is improper as 35 U.S.C. §102(b) requires that the printed publication or patent be published at least a year prior to filing. The prior art cited by the patent office is dated October, 2002. The present invention claims priority back

to September 2003. Applicant respectfully requests that the Patent Office withdraw the rejection.

Finally, the The Patent Office rejected Claims 47 and 48 under 35 U.S.C. §102(b) as being anticipated by *McKinsey*.

The Patent Office states that *McKinsey* teaches a method for discarding comparison instructions in a program, comprising: determining if there are any comparison instructions with discrete values in the program; and discarding a comparison instruction with a discrete value. The Patent Office further states that with respect to Claim 48, the prior art reference teaches a computer-readable medium storing computer-executable process steps of a process for discarding comparison instructions in a program, comprising: determining if there are any comparison instructions with discrete values in the program; and discarding a comparison instruction with a discrete value.

McKinsey teaches a method of compiling instructions of a program. The method includes receiving instructions for code motion and controlling the code motion while interacting with block ordering. The code motion may be done as part of various activities including instruction scheduling, partial redundancy elimination, and loop invariant removal. The scheduling may involve making an assessment of the cost of scheduling an instruction that takes into account generation and/or elimination of branches due to resulting block order update and determining whether to make the code motion based on the cost. Instruction scheduling may involve regeneration of predicate expressions to invert conditional branches.

Amended Claim 47 requires a method for discarding comparison instructions in a program, comprising: analyzing program instructions and tracking integer values; determining if there are any comparison instructions with discrete values in the program; discarding a comparison instruction and code outside of the determined discrete values and executing the program.

Amended Claim 48 requires a computer-readable medium storing computer-executable process steps of a process for discarding comparison instructions in a program, comprising: analyzing program instructions and tracking integer values; determining if there are any

comparison instructions with discrete values in the program; discarding a comparison instruction and code outside of the determined discrete values and executing the program.

McKinsey does not teach or suggest a method for discarding comparison instructions in a program, comprising: analyzing program instructions and tracking integer values; determining if there are any comparison instructions with discrete values in the program; discarding a comparison instruction and code outside of the determined discrete values and executing the program as required by Claim 47 and 48.

Under 35 U.S.C. §102(b), anticipation requires that a single reference disclose each and every element of Applicant's claimed invention. *Akzo N.V. v. U.S. International Trade Commission*, 808 F.2d 1471, 1479, 1 USPQ 2d 1241, 1245 (Fed. Cir. 1986).

Moreover, anticipation is not shown even if the differences between the claims and the reference are "insubstantial" and one skilled in the art could supply the missing elements. *Structure Rubber Products Co. v. Park Rubber Co.*, 749 F.2d. 707, 716, 223 USPQ 1264, 1270 (Fed. Cir. 1984).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (MPEP § 2131).

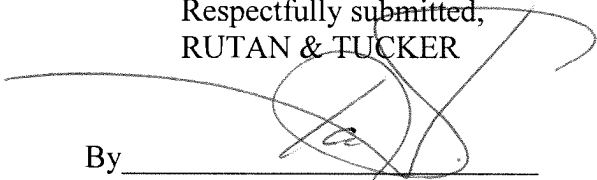
In view of the foregoing remarks and amendments, the rejection of Claims 47 and 48 under 35 U.S.C. §102(b) as being anticipated by *McKinsey* has been overcome and should be withdrawn. Notice to that effect is requested.

In view of the foregoing remarks, Applicant respectfully submits that all of the claims in the application are in allowable form and that the application is now in condition for allowance. If any outstanding issues remain, Applicant urges the Patent Office to telephone Applicant's attorney so that the same may be resolved and the application expedited to issue. Applicant requests the Patent Office to indicate all claims as allowable and to pass the application to issue.

Respectfully submitted,
RUTAN & TUCKER

Date: June 5, 2008

By



Hani Z. Sayed
Registration No. 52,544

Rutan & Tucker, LLP
611 Anton Blvd., 14th Floor
Costa Mesa, CA 92626-1931
Telephone (714) 641-5100
Fax (714) 546-9035